STM8 bootloader frequently asked questions

1 Introduction

All STM8A, STM8L, and STM8S devices with a Flash memory space greater than 16 Kbytes have a ROM bootloader:

- STM8AF51xx
- STM8AF61xx
- STM8AH61xx
- STM8S105xx
- STM8S207xx
- STM8S208xx
- STM8L151xx (using the USART)
- STM8L152xx (using the USART)

Please read the device datasheets for further information.

Please refer to the UM0500 (STM8A bootloader user manual) and the UM0560 (STM8L/S bootloader) for more information on how to use the respective bootloaders of the above families.

This document contains a list of frequently asked questions (FAQ) on the bootloader, which were received from users and subsequently answered and collected by customer support.

www.BDTIC.com/ST

Table of contents

1 Introduction 1

2 FAQ and answers 4

2.1 Which PC software can I use for connecting to the STM8 bootloader? . . . 4

2.2 Does PC software with command line support exist? 4

2.3 Which baudrate should I use with bootloader communication? 4

2.4 What should I do if there is no answer from the STM8 when I start the Flash loader demonstrator? 4

2.5 What should I do when the following message appears in the Flash loader demonstrator: Could not find E_W_ROUTINEs file for this version X.X. Please make sure you selected the right device ...? 5

2.6 How do I enable the bootloader? 6

2.7 How do I set the bootloader option byte through the Flash loader demonstrator? 6

2.8 How do I obtain the S19 file with the bootloader option byte inside it? . . . 7

2.9 What should I do if the device Flash memory cannot be programmed (stopping communication after write or erase commands are sent to the Flash) when I use my own uploading software? 8

2.10 Where do I obtain E_W routines for a device that I want to upload through the SPI (or UART) when using my own software? 8

2.11 What should I do if the following message appears when I want to connect to the device with the Flash loader demonstrator: Cannot get available commands, please try to change Echo selection, reset your device then try again...? 8

2.12 What does Echo mode mean? 9

2.13 When using SPI communication, what should I do if I have difficulties uploading the Flash memory or if the upload process stops? 9

2.14 What should I do if communication between the master and bootloader is not working when using CAN communication? 10

2.15 Is the bootloader active if readout protection (ROP) is enabled? 10

2.16 How can I upload firmware through the bootloader into an ROP-protected device? 10

2.17 Is an ROP-protected device safe from having its inside code read through the bootloader? 10

2.18 What is the solution for STM8 devices without a ROM bootloader? 11

www.BDTIC.com/ST

2.19 Other questions? 11

3 Revision history 12

www.BDTIC.com/ST

2 FAQ and answers

2.1 Which PC software can I use for connecting to the STM8 bootloader?

The PC software, Flash loader demonstrator, is used to connect to the STM8 UART via an RS232 port. It is provided by ST and can be downloaded free of charge from the ST website: *www.st.com*.

Note: The Flash loader demonstrator application is dedicated to the UART and LIN bootloader and is fully functional. However, it is provided only as a demonstration software. For custom applications, it is recommended that you write your own PC software according to the STM8A bootloader user manual (UM0500) or the STM8L/S bootloader (UM0560). Free PC software development tools can be used from the Internet (for example, GNU tools).

2.2 Does PC software with command line support exist?

Yes, the Flash loader demonstrator software has command line support. An STM8FlashLoader.exe program exits in the installation directory of the Flash loader demonstrator. This program is a command line version of the Flash loader demonstrator. Run it in MS-DOS command terminal, to obtain command line option descriptions.

2.3 Which baudrate should I use with bootloader communication?

The STM8 bootloader has automatic baudrate detection (when communication starts). Consequently, you can use baudrates from 4800 baud up to 500 kbaud.

2.4 What should I do if there is no answer from the STM8 when I start the Flash loader demonstrator?

Answer A

The bootloader must be enabled by the bootloader option bytes or the device has to be already erased.

Answer B

If the device is already programmed after reset, the bootloader waits for the host for only 1 s. Consequently, before connecting to the STM8 from the Flash loader demonstrator, you must reset the device and immediately press the “next” button in the PC software.

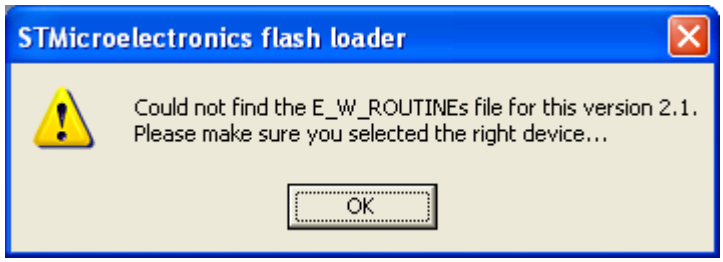
www.BDTIC.com/ST

2.5 What should I do when the following message appears in the Flash loader demonstrator: Could not find E_W_ROUTINEs file for this version X.X. Please make sure you selected the right device ...?

Answer A

In this case, the PC software has read the bootloader version X.X from the device and is trying to find the corresponding erase/write routines (E_W routines). Please confirm this message and select the given connected device from the drop down menu. Once the correct device is selected, this message will not appear again.

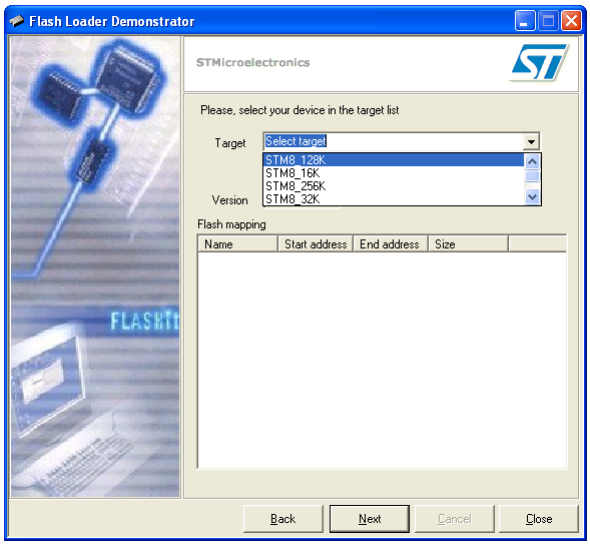
Figure 1. Dialog message when an E_W_ROUTINEs file for a specific bootloader version cannot be found


STMicroelectronics flash loader
Could not find the E_W_ROUTINEs file for this version 2.1.
Please make sure you selected the right device...
OK

Answer B

If the message appears again and you selected the correct device, a new version of the bootloader must exist inside the device for which there are no E_W routines in the PC software. Please update your PC software.

Figure 2. Dialog message when the PC software needs to be updated



www.BDTIC.com/ST

2.6 How do I enable the bootloader?

Answer A

If the device is erased, the bootloader is always active.

Answer B

If the device is not empty, the bootloader is activated by setting the bootloader option bytes to 55AAh. This can be done by STVP or manually by the SWIM interface. If the device is empty, it can be done by the bootloader itself. To enable the option bytes if the device is empty (bootloader active), write the correct bootloader option bytes to the device option byte area. As a consequence of bootloader option byte activation, the device Flash memory (which is now no longer empty) and the bootloader remain active up to 1 s after reset

Note: Enabling the bootloader (by setting the bootloader option bytes) is recommended as the first operation on virgin devices.

2.7 How do I set the bootloader option byte through the Flash loader demonstrator?

To set the bootloader option bytes, you must have an S19 file which contains a memory area with a bootloader option byte address inside (see example below of the .s19 with the option byte highlighted).

Example: Bootloader option byte for STM8S105x6 (address 487E and 487F)

```
S104480000B3
S104480100B2
S1044802FFB2
S104480300B0
S1044804FFB0
S104480500AE
S1044806FFAE
S104480700AC
S1044808FFAC
S104480900AA
S104480AFFAA
S104480B00A8
S104480CFFA8
S104480D00A6
S104487E55E0    <-          "Bootloader enable" option
S104487FAAE0    <-          Complemented "Bootloader enable" option
S9030000FC
```

Table 1. Reminder .S19 format

[record type]	[length_Add_and_CS_include]	[Add]	[data]	[CS]
S-	2bytes	4bytes	Nbytes	2bytes

www.BDTIC.com/ST

The Flash loader demonstrator offers *.S19 files for upload to the device. Ensure that this file has the correct bootloader option byte content and upload it onto the device. The bootloader automatically performs option byte programming (option byte area programming detection according to the addresses in *.S19 file).

Note: The address of the bootloader option byte may be different according to the chosen MCU. Please refer to the component datasheets.

2.8 How do I obtain the S19 file with the bootloader option byte inside it?

There are several ways to obtain the S19 file:

1. Use the Flash loader demonstrator
   a) Download the option byte area from the device to the S19 file.
   b) Change the bootloader option byte address for the correct value (55AAh) in the given saved file using an S19/HEX editor.
   Note: Also change the checksum in the edited S19 file line (see S19 file specification).
2. Use STVD in debug mode
   a) Start a simple STVD project with a given device.
   b) Connect the device and the SWIM.
   c) Start debug mode and set the correct value of the option bytes in STVD (using the *Debug instruments – MCU configuration* menu) to enable the bootloader.
   d) Open the memory window and right click on the mouse to open the *File – Save Layout* menu.
   e) Select the bootloader option byte addresses (for example 487Eh to 487Fh) and save to the S19 file.
3. Use STVP
   a) Start STVP and connect the device and SWIM.
   b) Go to the option byte page.
   c) Set the correct bootloader option byte value (to enable it).
   d) Save the option byte addresses to the S19 file.
   Note: If there is a bug, complementary bytes will not be saved. The bug should be removed, otherwise save to the S19 file through method (1) or (2).

www.BDTIC.com/ST

2.9 What should I do if the device Flash memory cannot be programmed (stopping communication after write or erase commands are sent to the Flash) when I use my own uploading software?

Before uploading content into the Flash/EEPROM memory you must firstly write E_W routines into the RAM. These RAM routines are called by the bootloader when the destination address is in the Flash or EEPROM memory.

Note: The Flash/EEPROM memory must be programmed from the RAM.

Next, upload E_W routines into the RAM address, A0h, using the WRITE bootloader command.

Note: STM8L devices do not require E_W routine downloading because the routines are stored inside device (not externally).

2.10 Where do I obtain E_W routines for a device that I want to upload through the SPI (or UART) when using my own software?

E_W routines are disseminated, free of charge, as attachment to the UM0560 and UM0500 user manuals on *www.st.com*. Routines are also present in Flash loader demonstrator installation subdirectory: \STM8_Routines. Load the appropriate routine to the RAM address, A0h, using the WRITE bootloader command.

2.11 What should I do if the following message appears when I want to connect to the device with the Flash loader demonstrator: Cannot get available commands, please try to change Echo selection, reset your device then try again...?

In this case, Echo mode probably needs to be enabled on the UART. Try to enable Echo mode in the PC software and to connect again. Echo mode can be enabled on the UARTs of the following devices:

- STM8S207/208 on UART3
- STM8S105 on UART2
- STM8A on LINUART

www.BDTIC.com/ST

Figure 3. Dialog message if experiencing problems when trying to connect to the device with the Flash loader demonstrator

STMicroelectronics flash loader
Cannot get available commands,
Please, Try to change Echo selection ,
Reset your device then try again...
OK

2.12 What does Echo mode mean?

Answer A

Echo mode is where each received byte from the device is sent back to the device. It is designed for a physical layer where the RxD and TxD lines are shared, providing a half duplex system with communication in both directions (via a LIN or RS485 connection).

Answer B

Echo mode can be enabled on UARTs (see *Section 2.11*)

Note: Normal mode (without Echo) is enabled on UART1 on the STM8S207/208 and on the USART on the STM8A51xx.

2.13 When using SPI communication, what should I do if I have difficulties uploading the Flash memory or if the upload process stops?

Answer A

SPI mode is master driven where the host pulls answers from device. It is therefore important to implement delays of adequate length on the host side when the host wants to obtain answers from the device. The device must be ready: sent commands have to be processed and the device must be prepared to send them to the host. See the UM0500 and UM0560 for the required delay calculations.

Answer B

An alternative to implementing long delays on the host side, is to use special E_W RAM routines which send back the “busy” status if the device is not ready. In other words, the host continues to poll the device until the “busy” byte answer disappears. This speeds up the upload process and provides safer implementation. Contact ST support to obtain the E_W RAM routines. See the UM0560 for further explanations.

www.BDTIC.com/ST

2.14 What should I do if communication between the master and bootloader is not working when using CAN communication?

A bug, due to unitialized CAN message filters, exists in STM8S208xx implementation of the CAN interface bootloader. Consequently, the user is unable to use the CAN interface with the bootloader in virgin devices. To solve this bug, the CAN message filters must be initialized from the user firmware followed by a jump to the bootloader (to address 0x6000). However, the user firmware must first be loaded into the device by the SWIM interface. The CAN bootloader works on this second call from the user firmware. ST can provide prepared user code which is locked in the user boot code area (UBC) from address 0x8000 to 0x8400. This code corrects the bug and the user firmware can start from address 0x8400 instead of 0x8000.

2.15 Is the bootloader active if readout protection (ROP) is enabled?

No, the bootloader does not run if ROP is set. This is to protect reading and/or writing code into the device (which protects the memory against Trojan horse malfunctions).

2.16 How can I upload firmware through the bootloader into an ROP-protected device?

Answer A

The bootloader is not active when ROP is set. However, the user application can jump back to the bootloader to a special bootloader entry point (after ROP check).

The jump back to the bootloader in ROP-protected devices is application dependent. In the user application, it is strongly recommended to follow an authentication process and implement a password check before allowing the user firmware to jump back to the bootloader.

Answer B

This situation is explained in the UM0560 user manuals.

2.17 Is an ROP-protected device safe from having its inside code read through the bootloader?

Yes, the device is safe from code reading because after startup the bootloader monitors ROP protection. If ROP is set, the bootloader jumps directly to the user application even if the bootloader has been enabled by the option bytes.

www.BDTIC.com/ST

2.18 What is the solution for STM8 devices without a ROM bootloader?

STM8 devices with a Flash memory below 16 Kbytes usually have no built-in ROM bootloader. However, the user can implement his own bootloader code (user-bootloader) and locate it in the UBC area. This area is located at the beginning of the Flash memory and can be write protected. It is highly recommended that the implemented code follows a similar structure as the built-in bootloader. However, the implemented code must also occupy some place in the RAM.

ST provides an example of the user bootloader which works the same way as the ROM bootloader. See the attached source codes in the AN2659. The user can modify source codes to fit them to a given STM8 type, to reduce space or add functionality.

2.19 Other questions?

For all other questions, please contact ST STM8 bootloader support or refer to the “mcu” web site on *www.st.com*.

www.BDTIC.com/ST

3 Revision history

Table 2. Document revision history

Date	Revision	Changes
19-Feb-2010	1	Initial release
17-May-2011	2	Replaced *Section 2.14: What should I do if communication between the master and bootloader is not working when using CAN communication?* Replaced “Flash memory” by RAM in *Section 2.18: What is the solution for STM8 devices without a ROM bootloader?*

www.BDTIC.com/ST

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER’S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2011 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com

www.BDTIC.com/ST